Rendezvous Search on the Edges of Vertex-Transitive Solids

by

Elanor West

A thesis submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Master of Science in Engineering

Baltimore, Maryland

December, 2018

Abstract

A classic "rendezvous search" problem is the "astronaut problem," in which two agents are placed on a sphere and move around until they meet. Research focuses on finding an optimal strategy for both agents to use. We consider a model that utilizes discrete units of time, with movement along the edges of vertex-transitive solids. The search ends when the two agents can see each other. We first examine the five platonic solids, then look at several larger Archimedean solids for comparison. We compare the mean times to meet on the solids under an unbiased random walk strategy, and we alter assumptions and strategies in various versions of the search to see how certain changes affect the mean time to end. One version involves the possibility of waiting on any given turn under both biased and unbiased random strategies. We also examine multi-step strategies, which involve a random step and a predetermined sequence of directions. The calculations of expected meeting times all involve first-step Markov chain decompositions.

Thesis Committee

Primary Reader

John C. Wierman (Advisor)
Professor
Department of Applied Mathematics and Statistics
Whiting School of Engineering
Johns Hopkins University

Acknowledgments

The research presented in this paper began in the course Introduction to Research in Discrete Probability, taught by Professor John C. Wierman in the spring semester of 2017. I would like to thank John C. Wierman for his inspiration and support over the last two years throughout the entire project. I would also like to thank Xiao (Annie) Xie, my partner throughout the whole project. Many of the baseline calculations presented were done in conjunction with Xiao Xie and most of Chapter 6 was investigated by her. This paper stems from a paper that the three of us submitted, and the information has been presented multiple times at national mathematics conferences. I would also like to thank Jeffrey Braun for presenting the open problem that inspired us to pursue this project and his research on closely-related problems. I thank Anan (Jasmine) Zhang for her help in the beginning with some initial calculations. I would like to thank the entire class from spring 2017, especially those not yet mentioned who supported us at multiple conferences: Yunfan Zhao, Gabrielle Moss, and Thomas Marge. I thank Professor Edward Scheinerman for some inspiring suggestions that helped motivate later research. I would like to thank the Acheson J. Duncan Fund for the Advancement of Research in Statistics for supporting me throughout my research, and Sandra Kirt for helping this process run smoothly.

Lastly, I would like to thank the entire Applied Mathematics and Statistics Department at Johns Hopkins University for the opportunity and support these past two years.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

Search theory involves an agent, or agents, looking for targets. These targets can be mobile or immobile, and the targets could be other agents. Search theory involves optimizing strategies for the agents to take during their search. But what do we mean by optimal? In this paper, we take optimal to mean the minimum expected time for the search to end. A branch of search theory that has been gaining attention in recent years is "rendezvous search." The premise of rendezvous search problems is that two (or more) agents are in different locations, and they want to find each other.

Search theory first came to light in the mid-twentieth century, during World War II. Many search problems stem from military operations. In 1975, Lawrence Stone obtained results involving a single agent looking for an immobile target [12]. This inspired others to look into a wider spectrum of search problems, particularly those involving a searching agent, with the goal of minimizing the time of the search, and another hiding agent, with the goal of maximizing the time of the search, much like a game of hide and seek [6]. In this paper, we focus on rendezvous search, which can be further broken down into two categories:

asymmetric and symmetric. In the asymmetric case, the players are not bound to the same strategy; in the symmetric case, they are. Work on the asymmetric case for n discrete locations has found that a "Wait for Mommy" strategy is optimal, in which one agent stays in place while the other searches all locations for the stationary agent. [5]. Steve Alpern, a pioneer in the field of rendezvous search, proposed ten open rendezvous search problems in Chapter 14 of *Search Theory: A Game Theoretic Perspective* [7]. Our research is inspired by the open problem titled the "astronaut problem."

The astronaut problem is as follows: two astronauts randomly land on a spherical planet, each with the same detection range in which they can see each other. Their goal is to find each other as fast as possible. No significant progress has been made on this problem. However, there are some results for rendezvous search on graphs and networks, which could potentially be used to approximate the sphere. One notable result is the Anderson-Weber Strategy for rendezvous search on n discrete locations in a complete graph. In this strategy, the agents have a probability of staying in the same location for $n-1$ steps and a probability of visiting all other $n-1$ locations for $n-1$ steps. If one agent waits and the other moves, then the agents are guaranteed to find one another. Meanwhile, if both agents move, there is no such guarantee. If both players wait, then they will definitely not meet. This strategy drew inspiration from the optimal "Waiting for Mommy" strategy in the asymmetric case, and has been proven optimal for 2 locations and 3 locations [5, 14].

Furthermore, Alpern considered rendezvous search on simple graphs in 1999, and both symmetric and asymmetric strategies were explored [8]. Specific cases on cycle graphs and complete graphs were examined, using similar techniques to our results on the platonic solids since the solids can be represented as graphs. In 2005, both asymmetric and symmetric strategies were explored on planar lattices. Alpern and Baston explore many agents on infinite lattices, finding optimal strategies for a few possible starting positions, but in the end propose that there are not necessarily optimal strategies in all starting positions [2]. More recently, in 2006, Alpern and Baston began to explore rendezvous search in higher dimensions. Most of their strategies that they claim to be optimal are asymmetric [1].

In investigating the astronaut problem, we decided to first simplify and approximate this problem with the five platonic solids: the tetrahedron, the octahedron, the cube, the icosahedron, the dodecahedron. The platonic solids are all vertex-transitive and all of their faces are identical, thus the possible moves of an agent on any given turn are the same on any platonic solid. We then started looking at some even larger vertex-transitive solids: the rhombicosidodecahedron and the truncated icosahedron. Vertex-transitive solids that are not the five platonic solids are called Archimedean solids. The Archimedean solids are basically three-dimensional graphs that have similar properties to the two-dimensional Archimedean lattices. Despite still being vertex-transitive, the Archimedean solids do not have identical faces.

We use a discrete model where each agent can move one edge length in one unit of time. We hope to find strategies that can be adapted to the original astronaut problem on a sphere. Our intuition is that "larger" solids will more closely approximate the sphere, though it is still necessary to compare both "larger" and "smaller" solids. Our ultimate goal is to minimize the expected time for the agents to see each other.

Chapter 2 sets up our version of the problem by introducing assumptions and background information needed to perform the calculations. Chapter 3 introduces and discusses unbiased random walks on the edges of the five platonic solids. We calculate these expected times using first-step Markov chain decompositions or using geometric probability mass functions. We use these results as baseline results to compare with other strategies. Intuitively, an unbiased random walk strategy might be considered “mindless” and would be expected to give longer expected meeting times than strategies where the agents have specific steps, though we see later that this is not always the case. Then, in Chapter 4, we consider a waiting probability and compare these results to the baseline strategies. We consider an unbiased waiting probability and then calculate optimal waiting probabilities in a random walk. Chapter 5 introduces unbiased random walks on larger Archimedean solids. The purpose of this is to add more edges and vertices so we can potentially find a closer approximation to a sphere. We briefly discuss multi-step strategies on some of the solids in Chapter 6, with up to seven steps. Chapter 7 discusses and compares results, while questioning how to correctly interpret said results. In Chapter 8, we conclude by comparing all strategies we have explored alongside a discussion of optimality and applications.

Chapter 2

Background and Setup

In this chapter, we will discuss the main assumptions of our version of rendezvous search, along with important background information.

2.1 Search Constraints

First, we created our rendezvous search problem with the following constraints:

1. The agents begin on vertices and travel along edges.

2. They must travel along full edge lengths, and may only travel from an edge to an incident vertex, and from a vertex along an incident edge (i.e. they cannot jump).

3. The agents can move one edge length per unit of time.

4. The agents each have the same given detection range in which they can see each other.

5. The agents begin such that they cannot see each other.

6. The search ends when the agents see each other.

Additionally, we will only examine *symmetric* strategies in this paper. This means that both agents have the same strategy and their turns are simultaneous. Note that the agents having the same strategy does not imply that they will move in the same manner at each turn, for most strategies described in this paper are random.

2.2 Vertex-Transitive Solids

We model our search on vertex-transitive solids.

Definition: A graph is *vertex transitive* if the graph's automorphism group acts transitively on all vertices.

This means that the vertices can be relabeled such that the graph seems identical to the original graph, but the vertices actually have different labels. In simpler terms, *vertex transitivity* means that at any unlabeled vertex, the solid looks the same in all directions as it does from any other vertex.

We first model the search on the five *platonic solids*. Platonic solids are regular, three-dimensional polyhedrons. There are only five possible platonic solids. The characteristics of the five solids are summarized in the table below.

Solid	Face Shape	Vertices	Edges	Faces
Tetrahedron	Triangular	4	6	4
Octahedron	Triangular	6	12	8
Cube	Square	8	12	6
Icosahedron	Triangular	12	30	20
Dodecahedron	Pentagonal	20	30	12

Table 2.1: Characteristics of the Platonic Solids

The platonic solids are all vertex-transitive, edge-transitive, and face-transitive. This means that, in addition to looking the same from all vertices, the solids look the same from all edges and faces as well. There are only five possible solids of this form.

Next, we explore two of the *Archimedean solids*, which are three-dimensional, vertex-transitive solids. However, these solids are not edge- or face-transitive, so the platonic solids are not considered Archimedean solids. These solids are typically *larger* than the platonic solids, and the Archimedean solids considered in this paper are much larger.

Definition: A three-dimensional solid is considered *larger* than another solid if its volume is greater when its edges are the same length. Additionally, a solid is considered *smaller* when its volume is less than another solids when both have equal edge lengths.

2.3 Detection Ranges

In our search, there are two possible detection ranges: full-face visibility and adjacent-vertex visibility.

Definition: An agent has *full-face visibility* if they can see all of the vertices belonging to the faces that include their current vertex.

Full-face visibility makes intuitive sense on three-dimensional solids because the agents can see everything on the planes that intersect at their position. They can see in every direction as far as the flat surfaces go.

Definition: An agent has *adjacent-vertex visibility* if they can see only vertices adjacent to their current vertex, meaning that there is one edge separating the agent from the vertices that they can see.

We get the following proposition given our two types of visibility on vertex-transitive solids.

Proposition: All faces on a vertex-transitive three-dimensional solid are triangular if and only if full-face visibility and adjacent-vertex visibility are equivalent.

Proof. We are given an agent on any node of a vertex-transitive, three-dimensional solid.

Forward Direction: We are given that all faces of the solid are triangular. Under full-face visibility, the agent can see all three vertices of each adjacent triangle. All of these vertices are either their current vertex, or one edge away from them. Additionally, the agent is able to see all vertices one edge away from them because they can see all adjacent triangular faces. Thus, the agent has adjacent-vertex visibility.

Backward Direction: We are given that full-face visibility and adjacent-vertex visibility are equivalent on a given solid. If the agent can see all vertices of all adjacent faces, then the faces must not have vertices greater than one edge away from the agent's current vertex. Since all of the faces are polygons, this means that all of these faces must have no two vertices separated by more than one edge. The only polygon for which this holds is the triangle. Thus, by vertex transitivity, all faces of the solid must be triangular. □

We define $T(solid)$ to represent the time that the search ends on the given solid. We will then define $E(T(solid))$ to represent the expected time for the search to end on the given solid throughout this paper. These quantities depend on the strategy used, so we will introduce subscripts to indicate the expected time for the search to end under each strategy.

Chapter 3

Unbiased Random Walk Strategy on the Platonic Solids

First, we will investigate an unbiased random walk on each of the five platonic solids. An unbiased random walk strategy is the simplest strategy on the edges of the solids. The two agents begin at vertices where they cannot see each other, and, at each iteration, move along a randomly chosen edge incident to their current vertex. By vertex transitivity, the agents have no preference for any direction, and thus even the first move is random. Thus, the agents do not have any sense of orientation at the start of the search. This strategy is equivalent to two simultaneous, unbiased, random walks on the edges of the platonic solid. The agents must move at each turn, and the search will end when the agents can see each other.

We investigate this strategy under both full-face visibility and adjacent-vertex visibility. Both versions of the unbiased random walk strategy provide upper bounds for the expected time it takes for an optimal strategy to result in the end of the search. Thus, we will use the following results as comparisons going forward when trying more strategic approaches.

3.1 Full-Face Visibility

We notice that on all five platonic solids, an agent can see at least half of the vertices of the solid from any given vertex under full-face visibility. On the five platonic solids, full-face visibility implies the following table.

Solid	Vertices	Visible Vertices	Fraction Visible
Tetrahedron	4	4	1
Octahedron	6	5	$\frac{5}{6}$
Cube	8	7	$\frac{7}{8}$
Icosahedron	12	6	$\frac{1}{2}$
Dodecahedron	20	10	$\frac{1}{2}$

Table 3.1: Full-Face Visibility: Visible Vertices

Calculations for this strategy can almost all be done using geometric probability mass functions, with the exception of the dodecahedron. This is due to the fact that on the smaller solids, if the agents cannot see each other, they are always in the same place relative to each other. Thus, each turn starts with the exact same orientation. On the dodecahedron, there are multiple ways for the agents to be oriented with respect to each other, so we use a first-step Markov chain decomposition to calculate expected time. We will also see that the icosahedron can be modeled with both a geometric probability mass function and a Markov chain representation. We will explore why this is the case, and compare the mean times for the search to end on all five platonic solids.

3.1.1 Tetrahedron

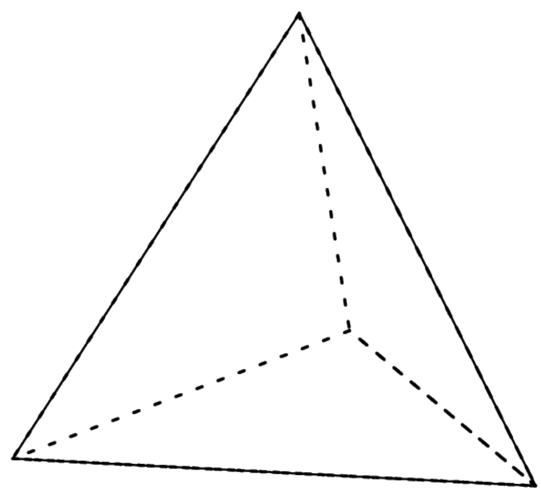

Figure 3.1: Tetrahedron

Any strategy with a detection range larger than zero is trivial on the tetrahedron. Since there are only four vertices, all of which are adjacent to each other, an agent on any vertex can see all other vertices. Thus, the expected time for the unbiased random walk strategy to end on the tetrahedron is zero under full-face visibility.

3.1.2 Octahedron

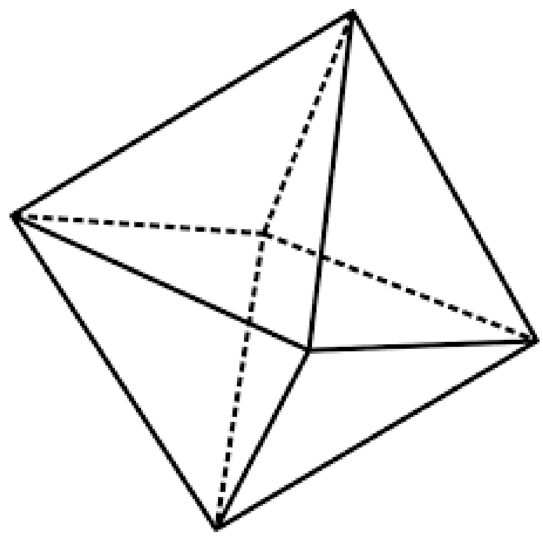

Figure 3.2: Octahedron

On the octahedron, each agent can see all but the vertex opposite its current vertex. Thus, if the search has not ended, any agent always knows where the other agent is, but it does not influence the randomness of its moves. Additionally, the four vertices that one agent can see aside from their current vertex are the same four vertices that the other agent can see aside from their current vertex. Each agent is going to move to one of those four vertices on the next turn. Thus, there is a $\frac{1}{4}$ probability of an agent going to any adjacent vertex on their next iteration. There is a $\frac{3}{4}$ probability that the agents will see each other after the first step. This is because there are only four possible vertices that the two agents could be on after one turn, and each agent will be able to see three of those four: their current vertex and the two adjacent to it that neither agent was on after the last turn. Consequently, given one agent's random choice of vertex, there is a $\frac{1}{4}$ probability that the two agents randomly choose the vertices exactly opposite each other.

To compute the expected meeting time, note that if the meeting time is exactly n units of time, the event that the agents did not see each other has happened $n-1$ times. Since each turn is independent from the last, the probabilities of these events can be multiplied together to get the probability of this happening, so it is $\frac{3}{4}(\frac{1}{4})^{n-1}$. Thus, we sum $n\frac{3}{4^n}$ over all possible n to get our expected time for the agents to see each other.

$$E(T(octahedron)) = 3\sum_{n=1}^{\infty} n(4^{-n}) = \frac{4}{3} \approx 1.3333.$$

3.1.3 Cube

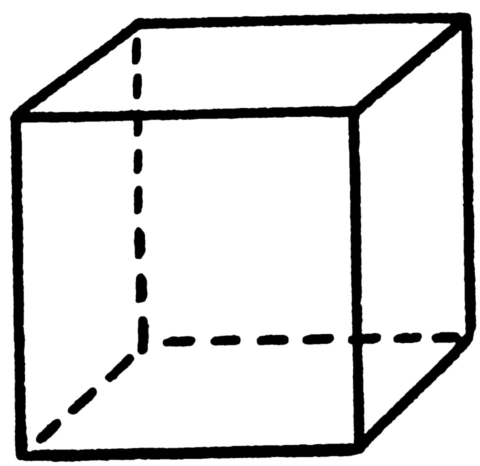

Figure 3.3: Cube

The setup for the cube looks very similar to that of the octahedron under full-face visibility. However, the agents could be up to two edges apart when they see each other. When the two agents see each other, it is plausible that they could be zero, one, or two edges apart. First, we need to look at their starting positions.

Proposition: The agents must begin three edges apart under the unbiased random walk strategy with full-face visibility on the cube.

We can easily see that this is true because the agents can see all vertices but one, which is the vertex opposite where they currently are, which is three edges away. So, the agents must be three edges apart. From this, we get the following corollary.

Corollary: The agents must see each other when they are one edge apart under the unbiased random walk strategy with full-face visibility on the cube.

Proof. The cube is a bipartite graph. So, the vertices can be partitioned into two independent sets of vertices. Any pair of vertices with an even number of edges between them must be in the same set, and any pair of vertices with an odd number of edges between them must be in different sets. So, the agents must start in different sets of the partition. At each iteration, the agents each move from one set to the other. Thus, the agents are never in the same set. Since the agents are never in the same set, they must never be an even distance apart. Since the agents can only see either one or two edges apart, this means they must see each other when they are one edge apart because they must be in different parts of the bipartition of vertices. □

Each agent has a $\frac{1}{3}$ probability of moving to any adjacent vertex on their turn, and there is a $\frac{2}{3}$ probability that the agents will see each other after any turn. Similarly to the octahedron, this is due to the fact that the agents must either be able to see each other, or be exactly opposite each other after one turn. Also like the octahedron, we noticed that the probability mass function is that of a geometric random variable.

$$E(T(cube)) = 2\sum_{n=1}^{\infty} n(3^{-n}) = \frac{3}{2} = 1.5.$$

3.1.4 Icosahedron

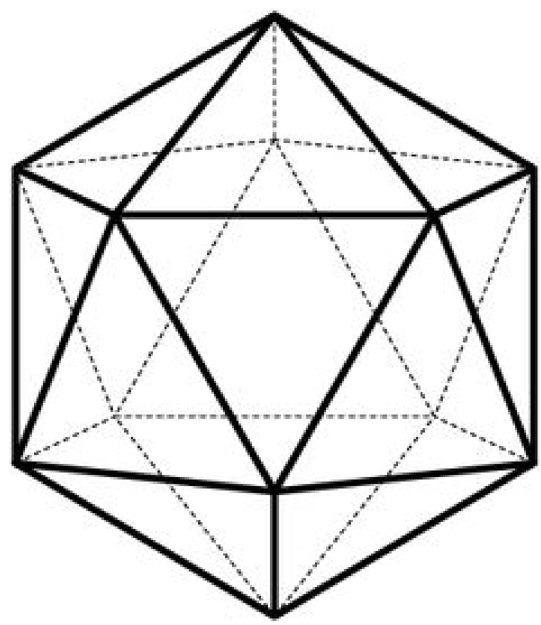

Figure 3.4: Icosahedron

The icosahedron is significantly larger than the previous three solids. Initially, the calculations for the icosahedron look similar to the previous solids because the expected time can be calculated using a geometric probability mass function. We discovered this initially when creating a table for each iteration. However, likely due to the fact that the agents can be either two or three edges apart and not see each other under this strategy, the sum appears to be more complicated than the previously examined solids.

$$E(T_(icosahedron)) = \frac{2}{5}\sum_{n=1}^{\infty}(\frac{3}{5})^{n-1}n = 2\frac{1}{2} = 2.5$$

On the icosahedron, the agents may begin either two or three edges apart. So, we can also show the calculations as a first-step Markov chain decomposition. This method is more intuitive than the geometric probability mass function due to the multiple starting positions. Let E_i denote the expected time to end if the agents start i edges apart, $i = 2, 3$.

The first equation below shows that there are $\frac{1}{6}$ and $\frac{5}{6}$ probabilities for the agents to begin three or two edges apart respectively. The second equation shows that, if the agents start two edges apart, they must make at least one iteration, and then they have a $\frac{2}{5}$ probability of being two edges apart again, a $\frac{13}{25}$ probability of being three edges apart, or they see each other. The third equation represents that, if the agents start three edges apart, they must perform at least one more iteration, with $\frac{1}{5}$ and $\frac{2}{5}$ probabilities of ending up three or two edges apart, respectively, or they see each other. We solve this system for the expected time for the search to end.

$$E(T(icosahedron)) = \frac{1}{6}E_3 + \frac{5}{6}E_2$$
$$E_2 = 1 + \frac{2}{5}E_3 + \frac{13}{25}E_2$$
$$E_3 = 1 + \frac{1}{5}E_3 + \frac{2}{5}E_2.$$

The system of equations can be algebraically manipulated to find the expected times given each starting position, and then those expected times are substituted into the top equation to find the overall expected time.

$$E_2 = \frac{5}{2},\ \ E_3 = \frac{5}{2}$$
$$E(T(icosahedron)) = \frac{5}{2} \approx 2.5.$$

As expected, both methods obtain the same results.

3.1.5 Dodecahedron

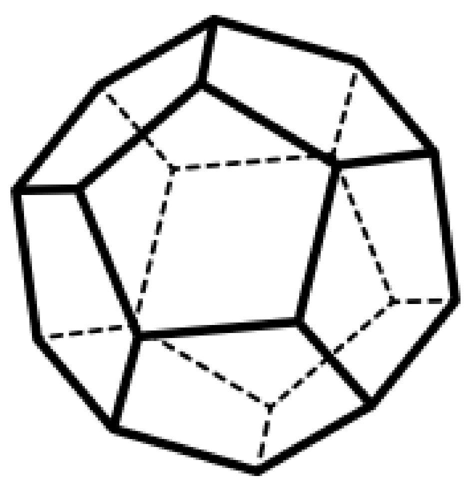

Figure 3.5: Dodecahedron

The dodecahedron calculations are set up similarly to the second set of icosahedron calculations. The agents have three possible starting positions on the dodecahedron: three, four, or five edges apart. Let E_i denote the expected time to end if the agents start i edges apart, $i = 3, 4, 5$.

The first equation represents the probability that the agents start three, four, or five edges apart, multiplied by the expected time for the search to end given the starting distance. The expected times for the search to end given the agents' starting distances are represented by the next three equations. Since we have four equations and four unknowns, we can algebraically solve the system.

$$E(T(dodecahedron)) = \frac{1}{10}E_5 + \frac{3}{10}E_4 + \frac{3}{5}E_3$$
$$E_3 = 1 + \frac{1}{9}E_5 + \frac{1}{9}E_4 + \frac{4}{9}E_3$$
$$E_4 = 1 + \frac{5}{9}E_4 + \frac{2}{9}E_3$$
$$E_5 = 1 + \frac{1}{3}E_5 + \frac{2}{3}E_3.$$

We briefly discuss the equations in more detail. The first equation represents the expected time for the agents to see each other. The coefficients are the conditional probabilities that the agents begin i edges apart, given that they do not see each other initially. The second equation represents the expected time for the agents to see each other if they are three edges apart. Since they cannot see each other, we will need to run at least one more iteration, hence the one. Then, there are nine possible ways for the agents to move, since they each have three options. Of these nine options: $\frac{1}{9}$ result in the agents becoming five edges apart, $\frac{1}{9}$ result in the agents becoming four edges apart, $\frac{4}{9}$ result in the agents becoming three edges apart again, and the other $\frac{1}{3}$ result in the agents becoming visible to each other, so these are multiplied by zero. Similarly, the third and fourth equations represent the expected time for the agents to see each other if they begin four or five edges apart, respectively. The coefficients in these equations also represent the probability that the agents become E_i edges apart after one iteration, given their starting positions.

Solving this system of equations, we get:

$$E_3 = \frac{51}{14}, \;\; E_4 = \frac{57}{14}, \;\; E_5 = \frac{36}{7}$$
$$E(T(dodecahedron)) = \frac{549}{140} \approx 3.9214.$$

3.1.6 Full-Face Visibility Expected Time Discussion

Solid	Face Shape	$E(T(solid))$
Tetrahedron	Triangular	N/A
Octahedron	Triangular	$\frac{4}{3} \approx 1.3333$
Cube	Square	$\frac{3}{2} = 1.5$
Icosahedron	Triangular	$\frac{5}{2} = 2.5$
Dodecahedron	Pentagonal	$\frac{549}{140} \approx 3.9214$

Table 3.2: Full-Face Visibility Summary

The table above summarizes the results for the expected time for the two agents to see each other when following an unbiased random walk strategy on the platonic solids. As one would predict, the larger solids have longer expected meeting times. The goal of our research is to eventually approximate the time for the search to end on the sphere. All of these solids can be represented on a sphere with vertices placed in such a way that the surface is similar to the solid, but curved. If we were to do this, however, the spheres would have to be different sizes if we are considering all edges to be of unit length. We will discuss possible ways of scaling these solids towards the end in Chapter 7, but it is important to keep this in mind. Right now, it is best to think of the expected time calculations as the expected number of turns until the search ends.

3.2 Adjacent-Vertex Visibility on the Platonic Solids

Now, we will look at the unbiased random walk with adjacent-vertex visibility on the five platonic solids. These detection ranges only differ on the cube and the dodecahedron, since their faces are not triangular.

Solid	Vertices	Visible Vertices	Fraction Visible
Tetrahedron	4	4	1
Octahedron	6	5	$\frac{5}{6}$
Cube	8	4	$\frac{1}{2}$
Icosahedron	12	6	$\frac{1}{2}$
Dodecahedron	20	4	$\frac{1}{5}$

Table 3.3: Adjacent-Vertex Visibility: Visible Vertices

As illustrated in the table above, this visibility restriction only changes the unbiased random walk strategy on two of the solids: the cube and the dodecahedron. This is due to their faces not being triangular, as discussed in Chapter 2.

The motivation for restricting visibility to a more uniform radius is that we could theoretically obtain a “better” approximation to the sphere, since an agent would be able to see just as far in all directions. On the sphere, the agents would be able to see the same distance in all directions, and the curvature of the sphere would be equal in all directions.

We will define the subscript A on the previously defined variable $E(T(solid))$ to denote results for adjacent vertex visibility on the given solid.

3.2.1 Cube

Here, we must solve with a first-step decomposition in a Markov chain, much like the icosahedron and dodecahedron's full-face visibility cases. This is because we now have two possible starting distances for the agents: two or three edges apart. We solve the following system of equations to obtain our expected time on the cube.

$$E(T_A(cube)) = \frac{1}{4}E_3 + \frac{3}{4}E_2$$
$$E_2 = 1 + \frac{2}{3}E_2$$
$$E_3 = 1 + \frac{1}{3}E_3.$$

Solving this system, we obtain:

$$E_2 = 3, \quad E_3 = \frac{3}{2}$$
$$E(T_A(cube)) = \frac{21}{8} = 2.625.$$

Intuitively, restricting visibility increased the expected meeting time from the full-face visibility version of the unbiased random walk. We can see that it takes longer for the agents to see each other when they start closer together. We suspect that this is due to the bipartite nature of the graph of the cube, as discussed in 3.1.3.

3.2.2 Dodecahedron

Limiting visibility on the dodecahedron yields calculations nearly identical to those with greater visibility, but with an added variable for being two-edges apart. The equations are:

$$
\begin{aligned}
E(T_A(dodecahedron)) &= \frac{1}{16}E_5 + \frac{3}{16}E_4 + \frac{3}{8}E_3 + \frac{3}{8}E_2 \\
E_2 &= 1 + \frac{1}{9}E_4 + \frac{2}{9}E_3 + \frac{1}{3}E_2 \\
E_3 &= 1 + \frac{1}{9}E_5 + \frac{1}{9}E_4 + \frac{4}{9}E_3 + \frac{2}{9}E_2 \\
E_4 &= 1 + \frac{5}{9}E_4 + \frac{2}{9}E_3 + \frac{2}{9}E_2 \\
E_5 &= 1 + \frac{1}{3}E_5 + \frac{2}{3}E_3.
\end{aligned}
$$

Solving this system of equations, we obtain:

$$
E_2 = \frac{285}{52},\ \ E_3 = \frac{393}{52},\ \ E_4 = \frac{114}{13},\ \ E_5 = \frac{471}{52},
$$
$$
E(T_A(dodecahedron)) = \frac{5907}{832} \approx 7.0998.
$$

Intuitively, the time for the search to end increases when the visibility decreases, just as on the cube. The times for the search to end when the agents begin four or five edges apart are very close, but it is still expected to be faster if they begin closer.

3.3 Adjacent-Vertex Visibility Expected Time Discussion

Solid	Face Shape	$E(T(solid))$	$E(T_A(solid))$
Tetrahedron	Triangular	N/A	N/A
Octahedron	Triangular	$\frac{4}{3} \approx 1.3333$	$\frac{4}{3} \approx 1.3333$
Cube	Square	$\frac{3}{2} = 1.5$	2.625
Icosahedron	Triangular	$\frac{5}{2} = 2.5$	$\frac{5}{2} = 2.5$
Dodecahedron	Pentagonal	3.9214	7.0998

Table 3.4: Adjacent-Vertex Visibility Comparison

The fractional values in the table above were computed by hand, whereas the decimal-only values were computed with MATLAB.

We can see from the table that, in addition to taking longer on the larger solids, it takes longer when the agents cannot see as far. This makes intuitive sense because each agent has less information at each turn when visibility is restricted. It is important to note that the increase in time due to restricting visibility is on the same solid, so it is proportionate to doing the same if the solid were drawn on a sphere. This is different than the increase in time due to the solids getting larger, because when all are drawn on a sphere, the vertices would no longer be unit distance apart on each solid's spherical representation. It is also interesting to note that restricting visibility on the cube makes the search longer than that on the icosahedron with full-face visibility.

3.4 Proof: Restricting Visibility Increases Expected Time

Based on the results presented in this chapter compared to the results in the previous chapter, we see that restricting the visibility of the agents leads to the search taking longer.

Remark: *Restricting* visibility implies that given a range of sight, the *restricted* visibility range is a subset of the original range of sight.

It is intuitive that taking away information from the agents will lead to a longer expected time. We will prove the following proposition.

Proposition: If visibility is restricted, then the expected time of any rendezvous search on a finite graph will increase.

When proving this proposition, we will use the subscripts F and A to denote two detection ranges such that F implies a larger detection range than A. Here, F can be thought of as full-face visibility and A can be thought of as adjacent-vertex visibility. We will use these subscripts while generalizing, though we only assume for the proof that the agents can see more vertices from their vertex under visibility F. This implies that the two visibilities cannot be equal.

Proof. Both agents are moving randomly at each turn. The agents have no intuition as to where the other agent is, other than that they are outside of their respective detection ranges.

Let $V(v) = \{$ Vertices visible to the agent at vertex v $\}$, with a subscript F denoting greater visibility than a subscript A.

We have that $V_A(v) \subset V_F(v)$. For example, we know that $(V_A(v) \backslash V_F(v)) = \varnothing$ if and only if the faces are all triangular under full-face and adjacent-vertex visibility.

Let agent 1 be at v_1 and agent 2 be at v_2. The agents will see each other under less visibility when $v_2 \in V_A(v_1)$ and $v_1 \in V_A(v_2)$. The agents will see each other under greater visibility when $v_2 \in V_F(v_1)$ and $v_1 \in V_F(v_2)$. Since $V_A(v) \subset V_F(v)$, we have that $v_2 \in V_A(v_1) \implies v_2 \in V_F(v_1)$ and $v_1 \in V_A(v_2) \implies v_1 \in V_F(v_2)$. Thus, if the agents can see each other under restricted visibility, then they can also see each other with greater visibility: $V_A(v_1) \cap V_A(v_2) \neq \varnothing \implies V_F(v_1) \cap V_F(v_2) \neq \varnothing$. This implies that the agents will take at least as long to see each other under restricted visibility as under greater visibility. Thus, we may conclude that restricting visibility implies that the search takes at least as long as under greater visibility, and possibly longer.

$\square$

Chapter 4

Permitting Waiting

In this chapter, we consider strategies that give the agents the option to wait at any iteration for a full unit of time instead of moving. In section 4.1, we consider an unbiased strategy, in which each agent's probability of waiting or traveling along any adjacent edge is uniform. For example, the waiting probability on the cube would be $\frac{1}{4}$ because there are three possible edges to travel, besides the option to wait. This strategy is still a form of an unbiased random walk strategy, just with an added option. We still consider it to be unbiased because the agent has the same probability of traveling along any incident edge or waiting. In section 4.2, we solve for an optimal probability of waiting on each solid given that the agents are still doing random walks. We will solve for optimal probabilities of waiting on the four nontrivial platonic solids and compare these results to the unbiased random walk strategies discussed in Chapter 3.

4.1 Unbiased Uniformly Distributed Waiting Probability

The calculations for the strategy with a uniformly distributed waiting probability are quite similar to those in our unbiased random walk strategy. However, the equation systems are more complicated and were solved using MATLAB. For each solid, we set up a first-step Markov chain decomposition, even for the solids where there is one possible relative position. We let ω denote the waiting probability. We then used the law of total probability to examine three cases: both agents move, both agents wait, and one agent moves while the other waits. Let W_1 be the event that the first agent waits and W_2 be the event that the second agent waits, at any turn, and let the superscript C denote complementation. We obtain the following equations.

$$P(W_1 \cap W_2) = P(W_1) \times P(W_2) = \omega^2$$

$$P(W_1 \cap W_2^C) = P(W_1) \times P(W_2^C) = P(W_1) \times P(1 - W_1) = \omega(1 - \omega)$$

$$P(W_2 \cap W_1^C) = P(W_2) \times P(W_1^C) = P(W_2) \times P(1 - W_2) = \omega(1 - \omega)$$

$$P(W_1^C \cap W_2^C) = P(W_1^C) \times P(W_2^C) = (1 - \omega)^2.$$

Let the subscripts WF and WA on T denote an expected time with a uniform waiting probability under full-face visibility or adjacent-vertex visibility, respectively.

4.1.1 Octahedron

The calculations for permitting a uniform waiting probability on the octahedron are similar to those without a waiting probability. There is one more option for the agents at each iteration, thus increasing the number of possible moves by one, and the four in the previous summand becomes a five. Similarly, all but one of these five possibilities leads to the agents seeing each other in one unit of time, thus increasing the coefficient to four from three. This is a result of the probability that the agents see each other at the end of one iteration increasing from $\frac{3}{4}$ to $\frac{4}{5}$. The new equation is shown below.

$$E(T_{WF}(octahedron)) = 4\sum_{n=1}^{\infty} n(5^{-n}) = \frac{5}{4} = 1.25.$$

4.1.2 Cube

4.1.2.1 Full-Face Visibility

The calculations for permitting a uniform waiting probability on the cube under full-face visibility are similar to those without a waiting probability. There is one more option for the agents at each iteration, thus increasing the number of possible moves by one, and the three in the previous summand becomes a four. Similarly, all but one of these four possibilities leads to the agents seeing each other in one unit of time, thus increasing the coefficient to three from two. This is a result of the probability that the agents see each other at the end of one iteration increasing from $\frac{2}{3}$ to $\frac{3}{4}$. The new equation is shown below.

$$E(T_{WF}(cube)) = 3\sum_{n=1}^{\infty} n(4^{-n}) = \frac{4}{3} \approx 1.3333.$$

4.1.2.2 Adjacent-Vertex Visibility

For adjacent-vertex visibility, the equations also change due to one more possibility at each iteration. The system of equations is as follows.

$$T_{WA}(cube) = \frac{1}{4}E_3 + \frac{3}{4}E_2$$
$$E_3 = 1 + \frac{4}{15}E_3 + \frac{2}{5}E_2$$
$$E_2 = 1 + \frac{2}{15}E_3 + \frac{7}{15}E_2.$$

Solving this system, we get:

$$E_2 \approx 2.5658, \; E_3 \approx 2.7632$$

$$E(T_{WA}(cube) = 2.6151.$$

4.1.3 Icosahedron

The system of equations for permitting uniform waiting on the icosahedron is below.

$$E(T_{WF}(icosahedron)) = \frac{1}{5}E_3 + \frac{4}{5}E_2$$
$$E_2 = 1 + \frac{1}{9}E_3 + \frac{1}{2}E_2$$
$$E_3 = 1 + \frac{1}{6}E_3 + \frac{5}{9}E_2.$$

Solving this system, we get:

$$E_2 \approx 2.6609, \; E_3 \approx 2.9739$$

$$E(T_{WF}(icosahedron)) \approx 2.7235.$$

4.1.4 Dodecahedron

4.1.4.1 Full-Face Visibility

The system for the dodecahedron under full-face visibility is below.

$$\begin{aligned} E(T_{WF}(dodecahedron)) &= \frac{1}{10}E_5 + \frac{3}{10}E_4 + \frac{3}{5}E_3 \\ E_3 &= 1 + \frac{1}{16}E_5 + \frac{3}{16}E_4 + \frac{7}{16}E_3 \\ E_4 &= 1 + \frac{1}{8}E_5 + \frac{3}{8}E_4 + \frac{3}{8}E_3 \\ E_5 &= 1 + \frac{1}{16}E_5 + \frac{3}{16}E_4 + \frac{7}{16}E_3. \end{aligned}$$

Solving this system of equations, we get:

$$E_3 \approx 4.2564,\ E_4 \approx 5.3846,\ E_5 \approx 6.1538$$

$$E(T_{WF}(dodecahedron)) \approx 4.7846.$$

4.1.4.2 Adjacent-Vertex Visibility

The system for the dodecahedron under adjacent-vertex visibility is below.

$$\begin{aligned} E(T_{WA}(dodecahedron)) &= \frac{1}{16}E_5 + \frac{3}{16}E_4 + \frac{3}{8}E_3 + \frac{3}{8}E_2 \\ E_2 &= 1 + \frac{1}{16}E_4 + \frac{1}{4}E_3 + \frac{3}{8}E_2 \\ E_3 &= 1 + \frac{1}{16}E_5 + \frac{3}{16}E_4 + \frac{7}{16}E_3 + \frac{1}{4}E_2 \\ E_4 &= 1 + \frac{1}{8}E_5 + \frac{3}{8}E_4 + \frac{3}{8}E_3 + \frac{1}{8}E_2 \\ E_5 &= 1 + \frac{1}{4}E_5 + \frac{3}{8}E_4 + \frac{3}{8}E_3 \end{aligned}$$

Solving this system of equations, we get:

$$E_2 \approx 6.5919,\ E_3 \approx 9.7074,\ E_4 \approx 11.0891,\ E_5 \approx 11.7316$$

$$E(T_{WA}(dodecahedron)) \approx 8.9247.$$

4.1.5 Summary and Discussion

Solid	ω	$E(T_W(solid))$	$E(T_{WA}(solid))$
Octahedron	$\frac{1}{5}$	$\frac{5}{4}$	$\frac{5}{4}$
Cube	$\frac{1}{4}$	$\frac{4}{3}$	2.6151
Icosahedron	$\frac{1}{6}$	2.7235	2.7235
Dodecahedron	$\frac{1}{4}$	4.7846	8.9247

Table 4.1: Uniform Waiting Probability Summary

The decimal values are approximations performed in MATLAB, while the fractional values are exact, and were computed by hand. For the octahedron and cube, the expected meeting time decreases from that of the unbiased random walk strategy.

On the octahedron, having one agent wait guarantees that the agents will see each other at the end of that iteration. If neither waits, it is the same result as both moving to vertices exactly opposite each other if they were to both move. There is only a $\frac{1}{25}$ probability that both agents wait and a $\frac{4}{25}$ probability that they both move but to opposite vertices. The probability that the two agents do not see each other at the end of any given turn decreases from $\frac{1}{4}$, under the unbiased random walk strategy, to $\frac{1}{5}$ when there is a uniform waiting probability. Thus, the expected meeting time will decrease slightly as well.

Results for the cube are similar to the octahedron. The probability that both agents wait is $\frac{1}{16}$ and the probability that they do not see each other if they do both move is $\frac{9}{16}$. Thus, under full-face visibility the probability that the agents do not see each other after any given turn also decreased from $\frac{1}{3}$ in the unbiased random walk to $\frac{1}{4}$ when there is a uniform waiting probability. Thus, the expected time for the cube also decreases. The reasoning for the adjacent-vertex visibility case on the cube is similar. From this, we can conclude that an optimal waiting probability on the octahedron and cube will be greater than zero. This is because the expected meeting times decreased when a uniform waiting probability was allowed. Thus, we know that this is faster than the unbiased random walk strategy without the option to wait, so permitting a probability of waiting decreases the expected time.

The expected times on the icosahedron and dodecahedron both increase, when compared to the unbiased random walk strategy. An intuitive explanation is that the agents cannot see the same vertices as each other. On both the octahedron and cube, the agents can see at least half of the vertices from any vertex. On the icosahedron and dodecahedron, the agents can see at most half of the vertices at a time. This means that the probability of an agent coming into view of the other waiting agent on a given iteration is much smaller, and depends on the initial distance between the two agents. Clearly, these solids are more complicated, and a higher probability of waiting farther away from the other agent will decrease the expected time for the agents to meet. What these results tell us is that when we calculate the optimal waiting probabilities, we know they will be less than the uniform waiting probabilities assigned in this section.

4.2 Optimal Waiting Probability

In this section, we calculate optimal waiting probabilities for the agents on any given iteration under the random walk strategy. We no longer say "unbiased" random walk strategy because the waiting probability may not be uniform, thus the random walk could be considered "biased" in the direction of the options with higher probabilities. We define *optimal waiting probability* as follows.

Definition: The *optimal waiting probability* for an agent performing a random walk on a platonic solid is the probability with which the agent should remain on their current vertex at any iteration to minimize the expected length of the search. Both agents will have the same optimal waiting probability, since we are working with symmetric strategies.

We know, from the previous section, that waiting can improve the search time on the octahedron and cube, and may or may not improve the search time on the icosahedron and dodecahedron.

To calculate the optimal probability of waiting, we must consider three cases: both agents wait, one agent waits, neither agent waits. We let ω_* denote the optimal waiting probability. These three scenarios have probabilities ω_*^2, $2(1-\omega_*)$, and $(1-\omega_*)^2$, respectively. To solve for the optimal waiting probability, we set up first-step Markov chain decompositions for all three scenarios, then combine them in the way described in the previous section. However, unlike in the calculations for the uniform waiting probability, we cannot substitute a value for the waiting probability.

We obtain a system of equations much like that in the calculations for the unbiased random walk strategy on the dodecahedron, but with coefficients for the E_k terms which are quadratic in ω_*.

Let the subscripts OW and OWA denote the expected search time with the optimal waiting probability under full-face visibility and adjacent-vertex visibility, respectively. Below, we have the linear program for the cube with adjacent-vertex visibility.

$$\text{Minimize:}$$

$$E(T_{OWA}(cube)) = \frac{1}{4}E_3 + \frac{3}{4}E_2$$

$$\text{subject to:}$$

$$0 \leq \omega_* \leq 1$$

$$E_3 = 1 + \frac{1}{3}(1-\omega_*)^2 E_3 + 2\omega_*(1-\omega_*)E_2 + \omega *^2 E_3$$

$$E_2 = 1 + \frac{4}{9}(1-\omega_*)^2 E_2 + \frac{2}{3}\omega_*(1-\omega_*)E_3 + \omega_*^2 E_2$$

We want to solve the system for all the expectations E_k. We do not have a new constraint, but we do have a new variable. We want to find the shortest possible time under which all constraints are satisfied, so we can set up linear programs to solve for ω_*. When we write-up the linear programs in MATLAB, we must define ω_* as a variable first, so we get our minimum expected time in terms of ω_*. Then, we substitute the vector of E_k terms into the objective function, and use basic calculus to minimize the function with respect to ω_* on the interval [0, 1]. Below are the results to the system presented above.

$$\omega_* = \frac{1}{4} = 0.25, \; E(T_{OWA}(cube)) = \frac{4}{3} \approx 1.3333.$$

The optimal probability of waiting on the cube turns out to be the uniform waiting probability. We find that this is also true on the octahedron and cube for both cases of visibility that we have studied. However, the icosahedron and dodecahedron yield different results.

The calculations for the icosahedron and dodecahedron have longer equations, and more of them. The process used was the same as for the octahedron and cube, although the results were drastically different. When minimizing the objective function with respect to ω_* over $[0, 1]$, MATLAB gives us very small numbers. They are very close to zero, but not exactly zero. After trying to increase the interval at which the function can minimize over, letting it include small, negative values, we still obtained the same results. All positive values of ω_* that were tested yielded longer search times than having no waiting probability at all. Thus, we conclude that, due to probable round-off error in MATLAB, the optimal waiting probabilities on the icosahedron and dodecahedron, under both full-face visibility and adjacent-vertex visibility, are all zero. MATLAB can only store numbers accurately up to fifteen decimal places, in a best-case scenario. Since the steps of the calculations involve both addition and subtraction of very small numbers, there is likely some round-off error in each step, which can add up fairly quickly. Thus, we end up with a large error term added to our optimal objective function value. This means that, if our optimal value is zero, then the result in MATLAB is actually the value of the round-off error. In conclusion, this means that the expected times for the search to end, if the agents use the optimal waiting probability, is the same as for the unbiased random walk strategy on the icosahedron and dodecahedron.

4.3 Waiting Probability Discussion

Solid	$\omega*$	$E(T_{OW})$	$E(T_{OWA})$
Octahedron	$\frac{1}{5}$	$\frac{5}{4}$	$\frac{5}{4}$
Cube	$\frac{1}{4}$	$\frac{4}{3}$	2.6151
Icosahedron	0	$\frac{5}{2}$	$\frac{5}{2}$
Dodecahedron	0	3.9214	7.0998

Table 4.2: Optimal Waiting Probability Summary

The table above represents the results of the calculations for the optimal waiting probabilities on all of the solids. The decimal values are approximations done in MATLAB. This table will look very familiar because the results are in the table of the results from the uniform waiting probability for the smaller solids and in the table for the unbiased random walk strategy (without waiting) for the larger solids.

Chapter 5

Unbiased Random Walk on Larger Archimedean Solids

Next, we will show the calculations for the unbiased random walk strategy on some larger Archimedean solids: the rhombicosidodecahedron and the truncated icosahedron. The calculations are done very similarly to those on the platonic solids, except the systems are significantly larger and MATLAB is necessary to compute all of the expected times.

For these larger solids, the faces are not all the same. This implies that there are multiple different relative positions of the agents in which they are the same number of edges apart. We will discuss what this means on each solid. Thus, the calculations presented in this chapter are technically approximations. The results are still interesting to compare, and we will discuss more of this throughout this chapter and in Chapter 7, when we discuss modeling onto the sphere. This chapter will discuss results by solid instead of by strategy, for ease of discussion, since the solids have more intricate differences and properties.

5.1 Rhombicosidodecahedron

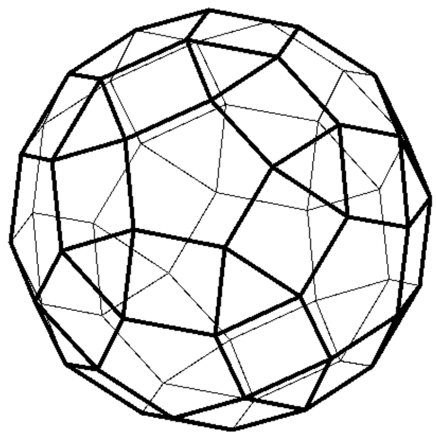

Figure 5.1: Rhombicosidodecahedron

The rhombicosidodecahedron has 60 vertices and 120 edges. It includes 12 pentagonal faces, 20 triangular faces, and 30 square faces. Each vertex is included in one pentagon, one triangle, and two squares. We explore both full-face and adjacent-node visibility using the unbiased random walk strategy. However, the calculations are approximations, rather than exact answers, due to the fact that the agents can be oriented differently, yet be the same distance apart. For instance, of the eight possible ways for the two to be oriented 2-apart (if one agent is fixed), four of them are distinct. Let $n \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ denote the distance between the two agents on the rhombicosidodecahedron and $N_n \in \mathbb{Z}^+$ denote the number of ways for the agents to be n edges apart if one agent is fixed. We find that on the rhombicosidodecahedron, if N_n is even, then there are $\frac{N_n}{2}$ distinct ways for the agents to be n edges apart, and if N_n is odd, then there are $\frac{N_n+1}{2}$ distinct ways for the agents to be n edges apart. This is because

$\lfloor \frac{N_n}{2} \rfloor$ of these orientations are symmetric, and if N_n is odd and there is one more, there is a unique orientation when the agents are n edges apart. The existence of at least 30 possible relative positions of the agents makes the exact calculations for the unbiased random walk strategy very difficult, especially when doing initial counting, because we would need an equation for each possible distinct orientation. We cannot code the systems into MATLAB until we have manually counted the possibilities for the agents to move and be oriented relative to each other. Thus, this is highly susceptible to human error.

5.1.1 Full-Face Visibility

Under full-face visibility, the agents can see up to two edges apart across the pentagonal face and the two square faces adjacent to them, and only one edge away across the triangle. Each agent can see a total of nine vertices, which is $\frac{3}{20}$ of the total vertices. The variable E_2 in the system below represents the expected time given that the agents are two edges apart and cannot see each other. We must clarify that since $N_2 = 8$ and the agents can see each other under 4 of these, we estimate the probability that the agents see each other at two edges apart is $\frac{1}{2}$. The system we solved to approximate the expected time is shown below. This system has been algebraically manipulated to have all constant terms on the right side of the equation, and all variables on the left.

$$E(T(rhombicosidod.)) = \frac{E_8 + 4E_7 + 8E_6 + 11E_5 + 12E_4 + 11E_3 + 4E_2}{51}$$

$$\frac{3}{16}E_4 + \frac{1}{4}E_3 - \frac{3}{16}E_2 = -1$$

$$\frac{9}{88}E_5 + \frac{21}{88}E_4 - \frac{13}{22}E_3 + \frac{1}{11}E_2 = -1$$
$$\frac{5}{48}E_6 + \frac{7}{32}E_5 - \frac{21}{32}E_4 + \frac{11}{48}E_3 + \frac{5}{96}E_2 = -1$$
$$\frac{1}{11}E_7 + \frac{2}{11}E_6 - \frac{27}{44}E_5 + \frac{5}{22}E_4 + \frac{5}{44}E_3 = -1$$
$$\frac{5}{64}E_8 + \frac{5}{32}E_7 - \frac{39}{64}E_6 + \frac{15}{64}E_5 + \frac{9}{64}E_4 = -1$$
$$\frac{1}{32}E_8 - \frac{15}{32}E_7 + \frac{3}{16}E_6 + \frac{1}{4}E_5 = -1$$
$$\frac{-3}{4}E_8 + \frac{1}{8}E_7 + \frac{5}{8}E_6 = -1.$$

Solving this system in MATLAB, we get:

$$E(T(rhombicosidodecahedron)) = 368.2010.$$

This is significantly longer than the smaller solids, but it is important to remember how much larger this solid is and how many more edges there are. This is where scaling the solids to have equal radii may help us compare more accurately, because all this number really tells us is the number of turns each agent must take. If we were to scale the solid down to have the same radius as a platonic solid, the edges would be much shorter, so it would take less real time to take each turn. We will discuss this further in Chapter 7.

As previously mentioned, these calculations are approximations. This is due to the multiple possible relative positions of the agents when they are a given distance apart. For example, when the agents are four edges apart, there are six possible relative orientations. The agents each have four possible directions to move in at the next iteration. This results in $(4 \times 4) \times (4 \times 4) \times 6 = 96$ possible relative movements of the agents on any iteration for which they begin four edges apart. After the agents move in one of these ninety-six ways, they can end up anywhere from two to six edges apart. This is represented in the following table.

Distance in Time $t+1$	Frequency	Probability
2	10	$\frac{5}{48}$
3	22	$\frac{11}{48}$
4	33	$\frac{11}{32}$
5	21	$\frac{7}{32}$
6	10	$\frac{5}{48}$

Table 5.1: Starting Distance: Four Edges Apart in Time t

We can see that the distribution is approximately symmetric around four, the starting distance. This is the case for most starting distances. The probabilities in the third column are where the coefficients in the system of equations come from. For example, these probabilities correspond to the fourth equation in the system above. The coefficients are nearly identical to these probabilities, but the coefficient for E_2 is half of this probability. This is because the agents are able to see each other under half of the possible ways for them to be two edges apart. Thus, the search would end if one of these distances occurred.

Where we have made approximations is in calculating the coefficients of the system. We treat all occurrences of E_4 as equal, meaning that this system is applied for all instances of E_4. However, it would be possible to compute, based on the starting distance, the exact probabilities for the agents to be a certain orientation of E_4 apart at a given iteration. This would not be possible to compute using a first-step Markov chain decomposition. We would end up with one equation in the system for each possible orientation, which includes one to eight for each possible distance apart.

The computation of a new linear system would not be mathematically challenging, but the absence of programs to compute these frequencies and probabilities is an obstacle. As of yet, all frequencies and probabilities must be computed by hand. The expected human error is so great that the calculations would very likely contain multiple errors. A program that would take the adjacency matrix or incidence matrix of the solid as input, and output the expected times for a random walk along the edges of the solid would be a tremendous aid in solving many versions of rendezvous search problems.

5.1.2 Adjacent-Vertex Visibility

Under adjacent-vertex visibility, the agents can only see five vertices, which is $\frac{1}{12}$ of the total vertices. This is slightly more than half of what the agents can see under full-face visibility. We make the same assumptions as under full-face visibility for adjacent-vertex visibility. This means we are still approximating the probabilities, thus the only difference between this system and the full-face visibility system is the value of E_2. Basically, we just double E_2 in each equation except the first. In the first equation, we add four to the denominator to account for the four new possibilities, and since we solved for E_2 initially, $-(1-\frac{3}{8})$ becomes $\frac{-5}{8}$ as the coefficient for E_2. Then, we plug this system into MATLAB. The system of equations is shown below.

$$E(T_A(rhombicosidod.)) = \frac{E_8 + 4E_7 + 8E_6 + 11E_5 + 12E_4 + 11E_3 + 8E_2}{55}$$

$$\frac{3}{16}E_4 + \frac{1}{4}E_3 - \frac{5}{8}E_2 = -1$$

$$\frac{9}{88}E_5 + \frac{21}{88}E_4 - \frac{13}{22}E_3 + \frac{2}{11}E_2 = -1$$
$$\frac{5}{48}E_6 + \frac{7}{32}E_5 - \frac{21}{32}E_4 + \frac{11}{48}E_3 + \frac{5}{48}E_2 = -1$$
$$\frac{1}{11}E_7 + \frac{2}{11}E_6 - \frac{27}{44}E_5 + \frac{5}{22}E_4 + \frac{5}{44}E_3 = -1$$
$$\frac{5}{64}E_8 + \frac{5}{32}E_7 - \frac{39}{64}E_6 + \frac{15}{64}E_5 + \frac{9}{64}E_4 = -1$$
$$\frac{1}{32}E_8 - \frac{15}{32}E_7 + \frac{3}{16}E_6 + \frac{1}{4}E_5 = -1$$
$$\frac{-3}{4}E_8 + \frac{1}{8}E_7 + \frac{5}{8}E_6 = -1.$$

Solving this system in MATLAB, we get:

$$E(T_A(rhombicosidodecahedron)) = 378.8987.$$

We see that these results are consistent with restricted visibility increasing the time of the search. It is expected to take nearly 11 more moves under adjacent-vertex visibility, though the expected time is relatively closer to the full-face visibility case on the rhombicosidodecahedron than on any of the platonic solids.

5.2 Truncated Icosahedron

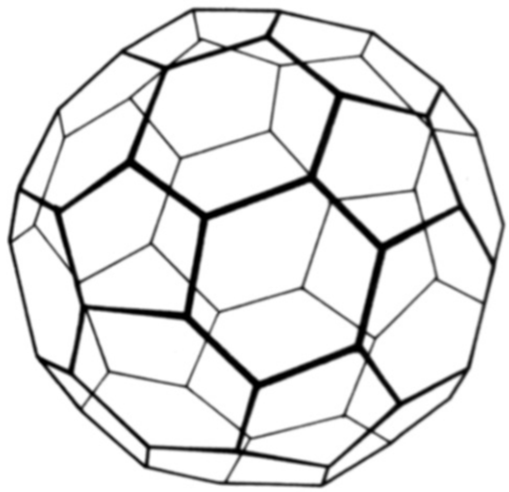

Figure 5.2: Truncated Icosahedron

The truncated icosahedron is better known as the *Buckyball*, the structure of the chemical *Buckminsterfullerene*, and the shape of a soccer ball. This shape is named after Richard Buckminster Fuller, an architect and inventor who introduced geodesic domes (think of the ball at Disney World in Epcot or the Montreal Biosphere) to modern-day architecture [10]. We explore this solid with the same motivation of more closely approximating the sphere, especially since this solid is widely accepted as a spherical approximation in society.

The truncated icosahedron has 60 vertices and 90 edges. There are 12 pentagonal faces and 20 hexagonal faces. Each vertex is included in two hexagons and one pentagon. The calculations we perform for this solid will be much like those for the rhombicosidodecahedron, in that they are approximations. This is due to the fact that the agents can be oriented differently and still be the same number of edges apart. For example, there are four possible ways for them to be three edges apart, and five possible ways for them to be five edges apart.

5.2.1 Full-Face Visibility

Under full-face visibility, the agents can see up to three edges away. They can see all vertices that are one or two edges away, due to the faces being pentagonal and hexagonal. At each vertex, the agents can see a total of 12 vertices including their current vertex. This is $\frac{1}{5}$ of the total vertices, which is greater than the visibility on the rhombicosidodecahedron. Additionally, the probability that the agents can see each other when they are three edges apart is $\frac{1}{4}$, so the E_3 coefficient is $\frac{3}{4}$ of the probability that the agents end up three edges apart. The system of equations we use to approximate the expected time is as follows.

$$T(truncated\ icosahedron) = \frac{E_9 + 3E_8 + 8E_7 + 10E_6 + 10E_5 + 10E_4 + 6E_3}{48}$$

$$\frac{7}{36}E_5 + \frac{1}{6}E_4 - \frac{5}{9}E_3 = -1$$
$$\frac{13}{19}E_6 + \frac{13}{90}E_5 - \frac{53}{90}E_4 + \frac{13}{120}E_3 = -1$$
$$\frac{2}{15}E_7 + \frac{8}{45}E_6 - \frac{3}{5}E_5 + \frac{2}{15}E_4 + \frac{7}{60}E_3 = -1$$
$$\frac{4}{45}E_8 + \frac{2}{9}E_7 - \frac{32}{45}E_6 + \frac{11}{45}E_5 + \frac{7}{45}E_4 = -1$$
$$\frac{1}{25}E_9 + \frac{1}{4}E_8 - \frac{13}{18}E_7 + \frac{5}{18}E_6 + \frac{1}{9}E_5 = -1$$
$$\frac{2}{75}E_9 - \frac{14}{27}E_8 + \frac{2}{9}E_7 + \frac{2}{9}E_6 = -1$$
$$-\frac{2}{3}E_9 + \frac{2}{9}E_8 + \frac{4}{9}E_7 = -1$$

Solving this system, we obtain:

$$E(T(truncated\ icosahedron)) = 19.9442$$

This is much longer than the unbiased random walk strategy on any of the platonic solids under full-face visibility, but much shorter than that on the rhombicosidodecahedron. This may be due to fact that there are $\frac{3}{4}$ as many edges, though this number still seems low relative to that of the rhombicosidodecahedron.

5.2.2 Adjacent-Vertex Visibility

Under adjacent-vertex visibility, the agents can see only four vertices, including their own. This is $\frac{1}{15}$ of the total vertices, which is $\frac{1}{3}$ of what they can see under full-face visibility. The difference between the system we obtain for adjacent-vertex visibility and full-face visibility is that the probabilities that the agents see each other when three edges apart decreases, and now the probability that they see each other when they are two edges becomes nonzero (i.e. it increases). The system of equations used to approximate the expected time is as follows:

$$T_A(truncated\ ico.) = \frac{E_9 + 3E_8 + 8E_7 + 10E_6 + 10E_5 + 10E_4 + 8E_3 + 6E_2}{56}$$

$$\frac{7}{27}E_4 + \frac{2}{27}E_3 - \frac{14}{27}E_2 = -1$$

$$\frac{7}{36}E_5 + \frac{1}{6}E_4 - \frac{17}{27}E_3 + \frac{2}{27}E_2 = -1$$

$$\frac{13}{19}E_6 + \frac{13}{90}E_5 - \frac{53}{90}E_4 + \frac{13}{90}E_3 + \frac{7}{45}E_2 = -1$$

$$\frac{2}{15}E_7 + \frac{8}{45}E_6 - \frac{3}{5}E_5 + \frac{2}{15}E_4 + \frac{7}{45}E_3 = -1$$

$$\frac{4}{45}E_8 + \frac{2}{9}E_7 - \frac{32}{45}E_6 + \frac{11}{45}E_5 + \frac{7}{45}E_4 = -1$$
$$\frac{1}{25}E_9 + \frac{1}{4}E_8 - \frac{13}{18}E_7 + \frac{5}{18}E_6 + \frac{1}{9}E_5 = -1$$
$$\frac{2}{75}E_9 - \frac{14}{27}E_8 + \frac{2}{9}E_7 + \frac{2}{9}E_6 = -1$$
$$-\frac{2}{3}E_9 + \frac{2}{9}E_8 + \frac{4}{9}E_7 = -1$$

Solving this system, we obtain:

$$E(T_A(truncated\ icosahedron)) = 32.3005$$

Given the results for full-face visibility, this result is intuitive. The expected time is longer than under full-face visibility. The result is still much smaller than the expected time for the search to end on the rhombicosidodecahedron, but is consistent with the previous calculations on the truncated icosahedron.

5.3 Relating Large Archimedean Solids to the Sphere

The results on the rhombicosidodecahedron and the truncated icosahedron give much longer expected times than the results on all of the platonic solids. This makes intuitive sense, due to more vertices, edges, and faces. Additionally, these solids would be much larger in volume than the platonic solids when all edges have the same length. Thus, the increase in time may be due in part to this increase. If the solids were scaled to have the same volume, the edges on larger solids would be shorter, so although more iterations would take place before the agents see each other, the iterations would take less time. We discuss this further in Chapter 7.

Chapter 6

Multi-Step Strategies

Next, we will look at a few multi-step strategies. A multi-step strategy is a symmetric strategy where both agents randomly choose a direction in which to move on their first turn, and then move in a predetermined n-step sequence consisting of steps in specific directions, creating an $n+1$-step strategy. In the strategies we will discuss, all of the other classic search regulations apply and the players have full-face visibility. The search ends whenever the agents see each other, including if they see each other in the middle of their sequence of movements. We again use a first-step decomposition Markov chain model to calculate the mean meeting time. Also, because of the vertex-transitivity of the platonic solids, opposite strategies yield the same expected times. This reduces the number of strategies that we need to analyze by a factor of 2.

This chapter does not include all details of the strategies mentioned, since most of the work on multi-step strategies is outlined by Xiao (Annie) Xie in her MSE Thesis, Johns Hopkins University Department of Applied Mathematics and Statistics, Spring 2019. This chapter touches on the elements of multi-step strategies that both of us were involved in and does not go into full detail.

6.1 The Two-Step Left Strategy

The simplest strategy we can construct is that the agents each first randomly choose a direction in which to move one step, and then move one step in the left direction relative to the edge that they just came from. We call this the Left Strategy. Recall that because of the vertex transitivity of the platonic solids, the expected times for opposite strategies are the same. So, the Left Strategy will yield the same expected time as the respective Right Strategy.

Let the subscript L denotes the Left Strategy:

The results for the octahedron, cube, and dodecahedron are as follows:

$$E(T_L(octahedron)) = 1.25.$$

$$E(T_L(cube)) \approx 1.3333.$$

$$E(T_L(dodecahedron)) \approx 3.2859.$$

The icosahedron allows for more possible two-step strategies because it also has the options of a hard left and a soft left (and respectively for the right). This is because each vertex has five adjacent vertices, so there are five possible edges to travel down on any turn. For any turn after the first, the agent can either go backwards, soft left, hard left, soft right, or hard right. Thus, we have a Hard Left Strategy where both agents first randomly choose a direction in which to move one step, and then move one step in the "hard left" direction relative to the edge they just came from, and a respective Soft Left Strategy.

Let the subscript HL refers to the Hard Left Strategy and SL refers to the Soft Left Strategy.

$$E(T_{HL}(icosahedron)) \approx 2.2921.$$

$$E(T_{SL}(icosahedron)) \approx 2.5273.$$

6.2 Optimality of the Left Strategy on the Octahedron and Cube

On the octahedron and cube, the Left Strategy has the property that the agents are guaranteed to see one another by the end of the second step. We were able to prove that the Left Strategy is optimal on both the Octahedron and Cube, and the proof is presented in Xiao Xie's thesis.

The general idea of both proofs is the same. The agents must start directly opposite each other on both the octahedron and cube. Then, the first step is random, so the agents either see each other, or end up exactly opposite again. Since the agents both must take a left if they do not see each other, they move towards each other. Then, on the octahedron, they must meet. On the cube, they must end up one-edge away, and they see each other. In both cases, the search ends in at most two iterations. This is proven to be optimal since the first iteration is always random, and the second, conditional on the outcome of the first, guarantees that they see each other.

Chapter 7

Spherical Comparison: Cube and Rhombicosidodecahedron

As previously mentioned, interpretation of the results presented in this paper may be better explained by scaling the radii of the solids. We consider the radius of a solid to be the distance from the center of the solid to a vertex, such that the center of the solid is equidistant to all vertices. This is called the *circumsphere radius*. Since the solids are vertex-transitive, there is a center that fits this criteria. This definition of radius may be less accurate to use to calculate approximations to the sphere than a radius equidistant to all faces, or the *midsphere radius*, but since the faces on the Archimedean solids are not the same shape, this radius may not be constant for a given solid.

To help visualize this, we will compare the cube and rhombicosidodecahedron, shown in the following figures.

It is clear that the cube's edges in the figure above are much longer than the edges of the rhombicosidodecahedron. Also, the volume of the cube in these figures appears to be larger than that of the rhombicosidodecahedron, so the figures are not a perfect comparison to size. We will use these two solids as an

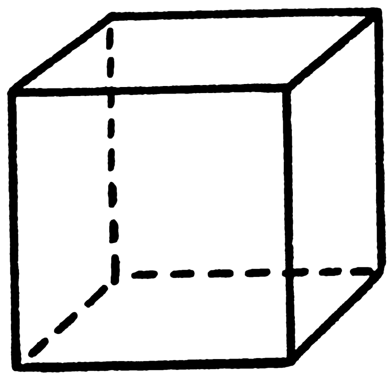

Figure 7.1: Cube for Scale

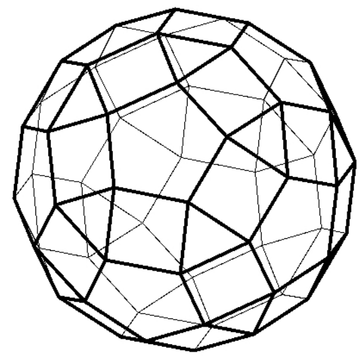

Figure 7.2: Rhombicosidodecahedron for Scale

example comparison for size and edge length.

We will set the volume of the rhombicosidodecahedron $\frac{4\pi}{3}$ units cubed, the volume of a sphere with radius 1. Then, the circumsphere radius of this rhombicosidodecahedron is 1.0387 units and the edges are 0.4652 units [13]. This radius is very close to that of a sphere with the same volume. Now, if we set the volume of the cube to be $\frac{4\pi}{3}$, we get that the circumsphere radius of the cube is 1.3960 units and the edges are 1.612 units [9].

Thus, we can see that a "larger" solid with the same volume as a "smaller" solid is a much better approximation to the sphere in terms of shape, due to the radius being closer to that of a sphere with the same volume. Additionally, with volumes still constant, the edges on the "larger" solid are significantly shorter, and in our case more than three times smaller, than those of the "smaller" solid. Per our example, if we scale the results for the expected times on the cube and rhombicosidodecahedron by the edge lengths, the new expected times for the search to end are outlined in the following table.

Solid	Edge Length	$E(T)$	$E(T_A)$
Cube	1.621	2.4315	4.2551
Rhombicosidodecahedron	0.4652	171.2871	176.2637

Table 7.1: Spherical Approximation

The expected time on the cube increases with this scaling of edges, due to the edge lengths increasing. Similarly, the expected time on the rhombicosidodecahedron decreases due to the edge lengths decreasing. Clearly, the search still takes many more iterations on the larger solid, but the time is cut nearly in half. On the cube, the time nearly doubles. So, we can see that the ratio of time on the rhombicosidodecahedron to that on the cube is decreased by a factor of nearly four. This may be intuitive, but it is important to keep in mind when comparing calculations on different solids. Similar comparisons could be done for all solids, and this example is meant to help explain the concept of scaling the edges to better compare the time for the search to end.

Another factor to consider given the increase in expected time would be to compare the proportion of vertices visible under the same visibility, but on a larger solid. We see that this likely affects the expected time on the platonic solids. We could think of this as the *effective detection range*. This range is larger on the smaller solids due to fewer total vertices.

Chapter 8

Conclusion

We have investigated expected meeting times for our simplified version of the astronaut problem on the platonic solids. We explored random walks, random walks with positive waiting probability, and multi-step strategies. As expected, there are intuitive patterns that appear, such as the expected times increasing when detection ranges decrease, and longer expected times on larger solids.

The unbiased random walk yields expected times that get longer as the solids get bigger. This makes sense because the area is larger. As visibility decreases from all adjacent faces to only adjacent vertices on the cube and dodecahedron, the expected time increases. This also makes sense because the agents can see less of the solid at any instant. These baseline cases are used as comparisons for other strategies, as our goal is to improve the expected meeting time. These results also give preliminary approximations for an upper bound to the astronaut problem, the search on the sphere.

When a positive waiting probability is added, the results are not consistent across solids. We find that it is optimal on the smaller, nontrivial solids, the octahedron and cube, to have a positive waiting probability. Additionally, we found that it is optimal to wait with a probability equal to that of going along any adjacent edge. The intuition behind this is that, under face visibility, the agents can see all but the vertex exactly opposite their current vertex, so if they cannot see each other, then they are exactly opposite each other. The addition of a waiting probability reduces the probability that the agents move to two exactly opposite vertices on their next turn. Interestingly enough, solving the system of equations for adjacent vertex visibility yields the same results on the cube.

On the larger platonic solids, the icosahedron and dodecahedron, the optimal waiting probability is zero. On these solids, the agents can see exactly half of the vertices of the solid at any time with face visibility. It makes sense that if the two agents move, they are more likely to see each other because it is more likely that they decrease, rather than increase, the distance between them. For the dodecahedron under adjacent vertex visibility, the intuition is the same. This result suggests that in the original astronaut problem on the sphere, the optimal waiting probability may be zero.

On the two large Archimedean solids that we examined, the rhombicosidodecahedron and the truncated icosahedron, the expected time for the unbiased random walk strategy increased greatly from that on the four non-trivial platonic solids. This is intuitive. However, the search takes much longer on the rhombisocidodecahedron than on the truncated icosahedron, although they have the same number of vertices, 60. The rhombicosidodecahedron does have more edges, 120 as opposed to 90. This is likely why the search takes longer on the rhombicosidodecahedron, since the agents have one more choice at each iteration than they do on the truncated icosahedron.

The Left Strategy serves as an introduction to multi-step strategies. It gives the agents a path that is not fully random. It decreases the expected time on the three solids examined, the octahedron, cube, and dodecahedron under full-face visibility. This verifies that non-random, multi-step strategies can be better than random strategies. In fact, we can prove that the Left Strategy is optimal on the octahedron and cube due to the fact that it guarantees the agents see each other by the end of the second step. It is important to note that the expected time for the Left Strategy on these smaller solids is equal to the expected time of the random walk strategy with the optimal waiting probability, but the random walk strategy could take an arbitrarily long time to end.

We began to look into how to scale the solids so that the volumes are equivalent. This gives insight to how closely each solid approximates the sphere, and how we may be able to interpret the expected time on each solid, since the solids are different sizes if the edges lengths are all of unit length. This concept could be explored further to potentially discover more accurate approximations on the sphere.

In additional and continuing research, we have briefly investigated an asymmetric alternating random strategy on the icosahedron, are currently looking at mixed multi-step strategies, have explored versions of the problem with no visibility, more Archimedean solids, and we are speculating how to possibly perform these calculations on solids that are not vertex-transitive. There are also many more asymmetric and symmetric strategies to be explored on vertex-transitive solids. Comparing these strategies will hopefully point us in the right direction for more complex approximations on the sphere.

Bibliography

[1] Alpern, Steve, and Baston, Vic. (2006) Rendezvous in higher dimensions. *SIAM Journal on Control and Optimization*, **44**, 2233–2252.

[2] Alpern, Steve, and Baston, Vic. (2005) Rendezvous on a planar lattice. *Operations Research*, **53**, 996–1006.

[3] Alpern, Steve. (1995) The rendezvous search problem. *SIAM Journal on Control and Optimization*, **33**, 673–683.

[4] Alpern, Steve, and Gal, Shmuel. (2011) *The Theory of Search Games and Rendezvous.* New York: Springer.

[5] Anderson, E. J., and Weber, R. R. (1990) The rendezvous problem on discrete locations. *Journal of Applied Probability*, **27**, 839–851.

[6] Gal, Shmuel. (1980) *Search Games.* Academic Press, New York.

[7] Alpern, Steve; Fokkink, Robert; Gasieniec, Leszek; Lindelauf, Roy; and Subrahmanian, V. S. (2015) *Search Theory: A Game Theoretic Perspective.* Springer.

[8] Alpern, Steve: Baston, V. J.; and Essegaier, Skander (1999) Rendezvous search on a graph. *Journal of Applied Probability*, **36**, 223–231.

[9] "Cube Calculator." *Rechner Online*, rechneronline.de/pi/cube-calculator.php.

[10] "Fuller, Arthur Buckminster." *New International Encyclopedia*. 1906.

[11] Stone, Lawrence D. (1989) OR Forum - What's happened in search theory since the 1975 Lanchester Prize? *Operations Research*, **37**, 501–506.

[12] Stone, Lawrence D (1975) *Theory of Optimal Search.* Academic Press, New York.

[13] "Rhombicosidodecahedron Calculator." *Rechner Online*, rechneronline.de/pi/rhombicosidodecahedron.php.

[14] Weber, Richard (2012) Optimal Symmetric rendezvous search on three locations. *Mathematics of Operations Research*, **37**, 111–122.

Elanor West
Baltimore, MD ewest13@jhu.edu 617 827 2496

Education

Johns Hopkins University **Baltimore, MD**

Master of Science in Engineering, Applied Mathematics & Statistics December 2018

Areas of Focus: Optimization and Operations Research, Discrete Mathematics

Bachelor of Science, Applied Mathematics & Statistics, French, and Economics May 2018

Professional Experience

Mathnasium **Baltimore, MD**

Math Instructor *May 2016 – Present*

- Teach math to children of all ages, kindergarten through AP Calculus, AP Statistics, AP Physics, and SAT/ACT prep
- Work in a classroom-like setting, similar to Montessori, where children of all ages work on different topics suited to their needs
- Previously worked at Mathnasium of Cohasset (Cohasset, MA) before Mathnasium of Roland Park (Baltimore, MD)

Thayer Academy **Braintree, MA**

Senior Camp Counselor *June 2012 – August 2016*

- Camp counselor for younger children, ages 3-7
- Taught children activities including art, field sports, swimming, kayaking, dance, and acting

North Atlantic Figure Skating Club **Falmouth, ME**

USFSA Basic Skills Coach *September 2010 – June 2014*

- Taught children the basics of figure skating
- Previously coached at the Skating Club of Hingham, MA, from September 2006 through June 2010

Research Experience and Presentations

JHU Department of Applied Mathematics and Statistics **Baltimore, MD**

Research Assistant *January 2017 – Present*

- Work under Professor John C. Wierman on the *astronaut problem*, a *rendezvous search* problem in which two players are placed randomly on a sphere and attempt to find each other. Currently working on mapping this problem on vertex-transitive 3-dimensional solids. There has been no previous progress on this problem.
- Funded by The Acheson J. Duncan Fund for the Advancement of Research in Statistics

Publications

West, E., Xie, X., Wierman, J. C. "Rendezvous Search on the Edges of Platonic Solids: Optimal Waiting Probabilities and Multi-Step Strategies." *Congressus Numerantium. (To appear.)*

Presentations

AMS Joint Mathematics Meetings 2019: *25 minutes, Baltimore, MD (Upcoming)* January 2019
MAA MathFest 2018: *15 minutes, Denver, C0* August 2018
JHU Engineering Design Day 2018: *Poster, Baltimore, MD* May 2018
JHU Day of Undergraduate Research (DREAMS): *Poster, Baltimore, MD* April 2018
Towson Regional Undergraduate Math Research Conference: *15 minutes, Towson, MD* April 2018
48th SEI Conference on Combinatorics, Graph Theory, and Computing: *15 minutes, Boca Raton, FL* March 2018
AMS Joint Mathematics Meetings 2018: *10 minutes, San Diego, CA* January 2018
MAA MathFest 2017: *15 minutes, Chicago, IL* July 2017

Honors and Awards

Pi Delta Phi, National French Honor Society: Xi Tau Chapter at Johns Hopkins University
Dean's List, Johns Hopkins University: Spring 2018, Fall 2017, Spring 2017
Outstanding Presentation Award: Awarded at MAA MathFest 2017 to the top 5%-10% of undergraduate presenters

Elanor West

Baltimore, MD ewest13@jhu.edu 617 827 2496

Leadership Experience and Involvement

Hopkins Figure Skating Club **Baltimore, MD**

Club Advisor *May 2018 - Present*

President *March 2017 – May 2018*

Vice President *January 2016 – March 2017*

- Organized and scheduled practices, fundraising, membership, and semi-annual performances
- Coached skaters of all levels
- Occasionally judged local USFSA Basic Skills Competitions

JHU Classical Ballet Company **Baltimore, MD**

Member *August 2014 – Present*

- Dance and perform with the company regularly, including at least semi-annual performances in ballet, modern, jazz, and contemporary style pieces

Alpha Phi Women's Fraternity, Zeta Omicron Chapter **Baltimore, MD**

Vice President of Risk Management *November 2016 – November 2017*

Director of Watchcare *November 2015 – November 2016*

- Participated in the Alpha Phi Eastern Leadership Conference twice: Newark, NJ, in February 2016 and Pittsburgh, PA, in February 2017
- Founded a Watchcare Committee that continues to help the chapter promote mental/physical health and safety
- Participated in the two highly-selective, five- and six-day intensive Emerging Leaders Institute and Leadership Fellows Programs at Butler University in Indianapolis, IN, in August 2016 and 2017, respectively. Improved leadership skills through group projects and discussions, professional workshops, and networking events
- Served on the Judicial Board and planned all social events for the chapter to ensure safety for everyone

Panhellenic Association at Johns Hopkins **Baltimore, MD**

Panhellenic Recruitment Guide *January 2018 – February 2018*

- Guided potential new members, mostly freshmen, through the Panhellenic Formal Recruitment process

JHU Modern Dance Company **Baltimore, MD**

Member *August 2014 – November 2016*

- Dance and performed with the company in our annual showcase each spring, all pieces were modern in style

Additional Skills

- Proficient in LaTeX, MATLAB, and Excel, and have exposure to Python, R, Stata, and Unix
- Professional working proficiency in French